

IMPLEMENTING WEB APPLICATION

PROJECT NAME	Skill / Job Recommender Application
TEAM ID	PNT2022TMID27679

Create the IBM Db2 service in the IBM cloud and connect the python code with DB

```
from flask import Flask,render_template,request,redirect
from flask_mysql import MySQL
import MySQLdb.cursors

app = Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
#MySQL username
app.config['MYSQL_USER'] = 'root'
#MySQL password here in my case password is null so i left empty
app.config['MYSQL_PASSWORD'] = 'Pragathi@01'
#Database name In my case database name is projectreporting
app.config['MYSQL_DB'] = 'sys'

mysql = MySQL(app)

@app.route('/joblist',methods=['GET','POST'])
def joblist():
    #creating variable for connection
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    #executing query
    cursor.execute("select * from candidate")
    #fetching all records from database
    data=cursor.fetchall()
    #returning back to projectlist.html with all records from MySQL which are
    stored in variable data
    return render_template("joblist.html",data=data)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

##### REGISTRATION #####

@app.route('/register')
def register():
    return render_template('register.html')
```

```

@app.route('/registerValidate',methods=['POST'])
def registerValidate():
    name=request.form.get('fullname')
    username=request.form.get('username')
    email=request.form.get('email')
    password=request.form.get('password')
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("""select * from `candidate` where `email` like '{}'
    """.format(email))
    ids = cursor.fetchall()
    if(len(ids)>0):
        return render_template('index.html')
    else:
        cursor.execute("insert into
candidate(name,username,email,password)values (%s,%s,%s,%s)", (name,username,em
ail,password))
        #fetching all records from database
        mysql.connection.commit()
        cursor.close()

        return render_template('registerValidate.html',name=name)

##### SIGN IN
#####

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/loginValidate',methods=['POST'])
def loginValidate():

    email=request.form.get('email')
    password=request.form.get('password')
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("""select * from `candidate` where `email` like '{}' AND
`password` like '{}' """ .format(email,password))
    duser = cursor.fetchall()
    if(len(duser)>0):
        return render_template('profilesetup.html')
    else:
        return render_template('login.html')

##### DashBoard
#####

@app.route('/jobposted',methods=['POST'])
def jobposted():

```

```

jtitle=request.form.get('jtitle')
location=request.form.get('location')
minyears=request.form.get('minyears')
jdes=request.form.get('jdes')
Category=request.form.get('Category')
salary=request.form.get('salary')
skills=request.form.get('skills')

cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("insert into
postjobs(jtitle,location,minyears,jdes,Category,salary,skills)values(%s,%s,%s
,%s,%s,%s,%s)",(jtitle,location,minyears,jdes,Category,salary,skills))
    mysql.connection.commit()
    cursor.close()
    return render_template('jobposted.html')

@app.route('/dashboard')
def dashboard():
    return render_template('dashboard.html')

@app.route('/users',methods=['POST'])
def user():
    fname=request.form.get('fname')
    lname=request.form.get('lname')
    currentaddress = request.form.get('currentaddress')
    permanentaddress = request.form.get('permanentaddress')
    mnumber = request.form.get('mnumber')
    alternatenumber = request.form.get('alternatenumber')
    email=request.form.get('email')
    alternateemail=request.form.get('alternateemail')
    cv=request.form.get('cv')
    photo=request.form.get('photo')
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("insert into
profilessetup(fname,lname,currentaddress,permanentaddress,mnumber,alternatenum
ber,email,alternateemail,cv,photo)values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(fna
me,lname,currentaddress,permanentaddress,mnumber,alternatenumber,email,altern
ateemail,cv,photo))
    #fetching all records from database
    mysql.connection.commit()
    cursor.close()
    return render_template('users.html')

##### Profile Setup
#####

```

```

@app.route('/profilesetup')
def profilesetup():
    return render_template('profilesetup.html')

@app.route('/completeSetup',methods=['POST'])
def completeSetup():
    fname=request.form.get('fname')
    lname=request.form.get('lname')
    currentaddress = request.form.get('currentaddress')
    permanentaddress = request.form.get('permanentaddress')
    mnumber = request.form.get('mnumber')
    alternatenumber = request.form.get('alternatenumber')
    email=request.form.get('email')
    alternateemail=request.form.get('alternateemail')
    cv=request.form.get('cv')
    photo=request.form.get('photo')
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("insert into
profilesetup(fname,lname,currentaddress,permanentaddress,mnumber,alternatenum
ber,email,alternateemail,cv,photo) values (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", (fna
me,lname,currentaddress,permanentaddress,mnumber,alternatenumber,email,altern
ateemail,cv,photo))
    #fetching all records from database
    mysql.connection.commit()
    cursor.close()

    return render_template('dashboard.html')

@app.route('/searchJob')
def searchJob():
    return render_template('searchJob.html')

@app.route('/postJob')
def postJob():
    return render_template('postJob.html')

@app.route('/sent')
def sent():
    return render_template('sent.html')

@app.route('/jobs',methods=['POST'])
def jobs():
    Category=request.form.get('Category')
    location=request.form.get('location')
    cursor=mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    cursor.execute("""select * from `postjobs` where `location` like '{}' AND
`Category` like '{}' """.format(location,Category))

```

```
data = cursor.fetchall()
if len(data)>0:
    return render_template('jobs.html',data=data)
else:
    return render_template('NoJobs.html')

@app.errorhandler(404)
def page_not_found(e):
    return render_template("404.html"),404

if __name__ == "__main__":
    app.run(debug=True)
```